

Práctica 2.3 Simulador de Von Neumann

jueves, 13 de octubre de 2022 13:34

1. Entra en la web <http://lab.xitrus.es/VonNeumann/>
2. Documenta explicando con tus propias palabras los diferentes **pasos** que se llevan a cabo en el *simulador* para al menos **3 de los programas** cargados y haz un cuadro resumen completando los siguientes datos:

OPERACION: $5 + 11 = 16$

Instrucciones utilizadas en el programa (operación)	Suma (código 0000, operación +)
El nº de instrucciones decodificadas	2 instrucciones decodificadas
nº de ciclos de CPU utilizados (aproximadamente)	8 ciclos de CPU
nº de accesos a memoria de lectura	2 accesos a memoria de lectura
nº de accesos a memoria de escritura	1 acceso a memoria de escritura

Instrucción "Sumar" (0000) decodificada:

El Decodificador interpreta los primeros 4 bits de la instrucción, que en este caso es el código 0000, correspondiente a la operación de sumar.

Ciclos de CPU:

Se utilizan **aproximadamente 8 ciclos de CPU** en total para ejecutar el proceso completo, ya que el flujo implica lectura de instrucciones, búsqueda de datos en memoria, envío al registro de entrada, ejecución de la operación y escritura del resultado.

Accesos a memoria:

Se realizan **2 accesos de lectura**: uno para la instrucción en sí y otro para los operandos (los números 5 y 11 que se van a sumar).

Se realiza **1 acceso de escritura** para almacenar el resultado (16) en el registro acumulador o en una celda de memoria.

PASOS

1. **Inicio:** El Contador de Programa (PC) envía su contenido al Registro de Direcciones (AR). El PC se incrementa para apuntar a la siguiente instrucción.
2. **Lectura de memoria:** La posición de memoria en la que se encuentra la instrucción de sumar se selecciona usando el AR, y la instrucción se envía al Registro de Datos (DR).
3. **Decodificación:** La instrucción se mueve al Registro de Instrucciones (IR), y el Decodificador lee los primeros 4 bits para interpretar que la operación a realizar es una suma (código 0000).
4. **Acceso a los operandos:** Los últimos 4 bits de la instrucción, que indican las direcciones de los operandos (en este caso, el 5 y el 11), se envían al Registro de Direcciones (AR) para buscar los valores en la memoria.
5. **Ejecución de la operación:** Los valores 5 y 11 se mueven al Registro de Datos (DR) y de ahí al Registro de Entrada (IN), donde el Circuito Operacional realiza la suma con el valor que estaba en el Registro Acumulador (AC), que en este caso inicialmente puede ser cero o uno de los operandos.
6. **Resultado:** El valor resultante, 16, se almacena en el Registro Acumulador o se escribe en una celda de memoria a través de un acceso de escritura.

OPERACION: $01001011 \text{ OR } 01010101 = 01011111$

Instrucciones utilizadas en el programa (operación)	OR (código 0101)
El nº de instrucciones decodificadas	2 instrucciones decodificadas
nº de ciclos de CPU utilizados (aproximadamente)	8 ciclos de CPU

nº de accesos a memoria de lectura	2 accesos a memoria de lectura
nº de accesos a memoria de escritura	1 acceso a memoria de escritura

Instrucción "OR" (0101) decodificada:

El Decodificador interpreta los primeros 4 bits de la instrucción, que en este caso es el código 0101, correspondiente a la operación OR (operación lógica entre bits que devuelve 1 si al menos uno de los bits es 1).

Ciclos de CPU:

Se utilizan **aproximadamente 8 ciclos de CPU** en total para llevar a cabo el proceso completo, lo que incluye lectura de instrucciones, acceso a operandos en memoria, ejecución de la operación OR y almacenamiento del resultado.

Accesos a memoria:

2 accesos de lectura: Uno para leer la instrucción (OR) y otro para leer los operandos (los valores 01001011 y 01010101).

1 acceso de escritura: Para almacenar el resultado de la operación OR (el valor 01011111).

PASOS

1. **Inicio:** El Contador de Programa (PC) envía la dirección de la instrucción al Registro de Direcciones (AR), el PC se incrementa para apuntar a la siguiente instrucción.
2. **Lectura de memoria:** Se selecciona la dirección indicada por el AR y se lee la instrucción, la cual se deposita en el Registro de Datos (DR).
3. **Decodificación:** La instrucción se transfiere al Registro de Instrucciones (IR), y el Decodificador interpreta los primeros 4 bits, identificando que la operación es **OR** (código 0101).
4. **Acceso a los operandos:** Los últimos 4 bits de la instrucción, que indican las direcciones de los dos números binarios a operar (01001011 y 01010101), son enviados al Registro de Direcciones (AR) para acceder a las celdas de memoria correspondientes y realizar la lectura de esos valores.
5. **Ejecución de la operación:**
 - Los números binarios se transfieren al Registro de Datos (DR) y de allí al Registro de Entrada (IN).
 - El Circuito Operacional realiza la operación **OR** entre los bits de ambos números, generando el resultado 01011111.
6. **Resultado:** El valor resultante 01011111 se almacena en el Registro Acumulador (AC) o en una celda de memoria a través de un acceso de escritura.

OPERACION $((2^2 + 2)^2 = 36$

Instrucciones utilizadas en el programa (operación)	Exponente (código 0100, operación ^)	Suma (código 0000, operación +)	TOTAL
El nº de instrucciones decodificadas	2 instrucciones decodificadas	1 instrucción decodificada	3 instrucciones
nº de ciclos de CPU utilizados (aproximadamente)	10 ciclos de CPU	4 ciclos de CPU	14 ciclos
nº de accesos a memoria de lectura	4 accesos a memoria de lectura	2 accesos a memoria de lectura	6 accesos
nº de accesos a memoria de escritura	2 accesos a memoria de escritura	1 acceso a memoria de escritura	3 accesos

1. Exponente 1: $2^2 = 4$

- **Instrucción decodificada:** código 0100 (exponente).
- **Accesos de memoria:**
 - **2 accesos de lectura:** uno para cada operando (2 y 2).
 - **1 acceso de escritura** para almacenar el resultado (4).

2. Suma: $4 + 2 = 6$

- **Instrucción decodificada:** código 0000 (suma).

- **Accesos de memoria:**
 - **2 accesos de lectura:** uno para cada operando (4 y 2).
 - **1 acceso de escritura** para almacenar el resultado (6).

3. Exponente 2: $6^2 = 36$

- **Instrucción decodificada:** código 0100 (exponente).
- **Accesos de memoria:**
 - **2 accesos de lectura:** uno para cada operando (6 y 2).
 - **1 acceso de escritura** para almacenar el resultado (36).

Ayúdate de la siguiente tabla de decodificación de las instrucciones utilizadas:

CODIGO	OPERACIÓN	SIMBOLO
0000	Sumar	+
0001	Restar	-
0010	Multiplicar	*
0011	Exponente	^
0100	AND	&
0101	OR	
0110	Mover a memoria	M
0111	Terminar	...